DESIGN TOOL FOR WIND TURBINE CONTROL ALGORITHMS

E.L. van der Hooft[1], vanderhooft@ecn.nl, Tel. +31 224 564913
T.G. van Engelen[1], vanengelen@ecn.nl, Tel. +31 224 564141
P. Schaak[1], schaak@ecn.nl, Tel. +31 224 564278
E.J. Wiggelinkhuizen[1], wiggelinkhuizen@ecn.nl, Tel. +31 224 564049

[1]ECN Wind Energy, P.O. Box 1, 1755 ZG Petten,
The Netherlands, Fax. +31 224 568214

ABSTRACT: Advanced wind turbine control algorithms have become more important over the last years in order to deal with high requirements on reliability, cost of energy and extreme operating (offshore) conditions. An open source modular 'Design tool for wind turbine control algorithms' within the Matlab environment enables possibilities for wind turbine designers to develop industrial control algorithms and to utilize the benefits of more advanced control solutions.

The design tool offers a proven design procedure, which takes the different design stages of a wind turbine into account. It supports initial design and evaluation of control algorithms, linking to aero-elastic codes and implementation in the turbine controller. In addition, the tool assists the designer to operate the design procedure, to avoid design failures and ordering of all the design data, models and versions.

Currently, the incorporated design and evaluation models are focussed on design of classic 'rotor speed feedback control' for a variable speed and active pitch turbine and have been verified in practice.

More advanced control design modules are within reach as a result of current developments on frequency domain analysis and synthesis of (linearised) turbine models.

1. INTRODUCTION

Advanced control solutions show good perspectives to improve and optimise wind turbine operation. They comprise high potentials for future turbine designs, to deal with different operation circumstances and to find a design optimum between costs and performance.

The objective of the research program 'Design tool for wind turbine control algorithms' as performed by ECN Wind Energy [10], is to

- make in-house design and evaluation of advanced control solutions more accessible for wind turbine manufacturers;
- stimulate the use of pertaining algorithms in modern turbines.

The dynamic behaviour is turbine specific and the footprint data of systems (like resonance's and design specifications) will usually be confidential.

First, a general overview on the design tool is given. More details are discussed in the next paragraphs (3, 4, 5) and finally, the benefits and main issues of the design tool are listed.

2. CONTROL DESIGN TOOL

The design tool has a modular structure and operates within the Matlab environment[1]. Basically, it needs two additional 'toolboxes' (signal processing toolbox and control toolbox) and, optionally, the graphical toolbox Simulink for time domain simulation.

Matlab enables powerful statements and quick programming possibilities due to its interpreter approach. The drawback of this approach is covered in the control design tool by using 'functions' (local workspaces) which are interconnected via unambiguous global structures. The use of the high level Matlab programming language enables the designer to modify design code or add new design modules easily ('open source approach').

The control design tool comprises the following features, which will be discussed in the next paragraphs:

- a proven design procedure;
- designer services;
- design and evaluation modules.

3. DESIGN PROCEDURE

In practice, the first control algorithms are often designed for load set calculations (certification) and, dependent of the results, a number of modifications will be made to both the turbine and control algorithm design.

A design procedure should be able to design and evaluate initial control algorithms before the turbine design data have been made definite and irreversible. Based on draft turbine specifications and pertaining footprints of the dynamics, an initial controller design and evaluation will reveal unfavourable constraints from viewpoint of control performance.

It will be also favourable to separate control algorithm designs and performance evaluations from (more detailed but time consuming) load evaluations as calculated by aero-elastic codes. Therefore, a seamless link of a control algorithm realisation to the aero-elastic code is necessary.

The design procedure as shown in figure 1, will take these requirements into account. It consists of three sequential *design activities*:

1. control design (3.1);
2. evaluation (3.2);
3. implementation (3.3).

In figure 1, the '*processes*' (circles) are connected by '*data flows*' (bold) via '*data stores*' (boxes). Dependent of the results of each activity ('sheet' shapes), it should be decided by the designer either to start the next activity or to iterate back into the procedure.

---

[1] The Matlab environment is an industrial accepted programming language of 'The Mathworks Inc.', for scientific calculations and analyses.

3.1 Control Design

The input for the first design activity is *design data* and intended *control targets.* The needed input data depends of the turbine concept, but concerns in general:

- aerodynamic data:
  - rotor power- and thrust coefficients or
  - lift and drag coefficients of blade profiles;
- structural parameters:
  - foundation/tower, drive train, rotor/blades;
  - resonance's, stiffness, dimensions, weights;
- requirements and constraints:
  - actuators and sensors;
  - nominal and extreme values turbine systems;
  - excitation sources (wind, waves).

Two main categories of data are defined, *turbine data* and *control design data*. Each category contains a collection of pertaining input files for related data.

The designer shall input the data mandatory into an *input data record* consisting of the parameter name, its value and meta-data like unit, description etc. Data can also be grouped to 'parent-child' notations: equal 'child' names are then permitted as long as they have different 'parents' (this is often used for state-space notations).

Design modules (Matlab m-code) processes the input data records to well structured and unambiguous data definitions in so called *'Design global structures*' (DGS) as required for the evaluation activity (3.2).

Design modules will also calculate additional data from input data. Redundant input data (inconsistencies) is herewith avoided. Most derived data definitions will be generated by:

- *control synthesis and analysis*, like dimensioning parameters for the intended control structures or sensitivities of linear design models;
- *simulation model dimensioning*, like non-linear model parameters or wind field data.

During the design activity, logging messages and verification plots are generated for observation, assessment and reporting purposes.

Based on (linear) control analysis within the design modules, the capability of actuators and sensors are proved, even as control loop stability and robustness.

If a result does not meet the expectations or requirements, the control design should be modified (control synthesis, parameter tuning) or the control targets should be adapted. If the results satisfy, the *control definition data* and the *turbine simulation data* will be stored in DGS's as a starting point for the next activity.

3.2 Evaluation

To evaluate the time domain control performance, the *control structure* and the *wind turbine model* as designed and dimensioned in the previous activity, should be implemented in a simulation kernel.

Both the *control structure* and the *wind turbine model* are implemented as simulation modules (Matlab m-code in a so-called s-function template). This allows the designer to use either the graphical time domain simulation toolbox (Simulink) or the build-in numerical integration scheme (Runge Kutta 4).

Using Simulink will be advantageous for demonstrations, while the build-in solver gives more sense towards realistic numerical implementation.

The sample rate at which each control module will be calculated can be different (tasks) as specified during the design activity. Obviously, the turbine dynamics will be calculated considerable faster in order to approximate 'time continuous behaviour'.

The design tool will facilitate in definition of the parameters and interacting signals, as required for the modules of the control structure and the turbine model.

The parameter definitions are mapped from the 'Design Global Structures' (DGS) to the so-called *'Evaluation Global Structures'* (EGS). Additional simulation related parameters are also defined within the EGS. EGS data can be either 'run-time-changeable' or not. *Run-time parameters* are mentioned for tuning or adjustment of preferred behaviour (limits, setpoints, weighting factors) and allowed to change in the evaluation activity without re-design.

The signals are mandatory defined by the designer as *input signal records* (similar to the *input data record* in the previous activity). The input and output ports of the modules are mapped together to create a time domain simulation model.

Before time domain simulation is ready to run, the acquisition of monitoring signals should be defined. Time series of these signals will be stored during simulation in related structures for each simulation module separately. After simulation, the design tool supports the designer in time domain assessment of the control design, by generation of time series plots.

If the time domain behaviour does not meet the expectations or requirements, there are two possibilities to modify:

- tuning of 'run-time-parameters' within the current evaluation activity;
- redesign the controller (*control synthesis and analysis*) by iteration to the previous control design activity.

If the results satisfy, the *control definition data* as stored in the EGS's are used as a starting point for the next activity.

3.3 Implementation

As soon as the results of the control design activity (3.1) and evaluation activity (3.2) are meeting the requirements and control targets, the control algorithm should be linked to an aero-elastic code for more detailed analysis towards fatigue and extreme turbine loading.

The design tool will facilitate this conversion to a high degree. The designer should only made minor modifications before the Matlab C-compiler will generate automatically a Dynamic link library (DLL) from the *control structure modules* and pertaining *control definition data:*

- prepare a calling main function to the control structure modules;
- provide a gateway towards the aero-elastic code.

To check if the conversion process has passed successfully, the design tool enables the possibility to run the DLL in parallel to the simulations as used during the evaluation activity (3.2). Obviously, differences larger than the numerical accuracy are not allowed.

After linking the control algorithm (DLL) to the aero-elastic code, the resulting control performance and turbine loading will be assessed in more detail with respect to the evaluation activity in 3.2.

The DLL will fetch the as 'run-time' defined parameters from an ASCII file. This enables the possibility to change the algorithm during 'load calculation' to a limited and safe extend.

If the performance does not meet the control targets or requirements, the same possibilities as mentioned in the evaluation activity (3.2) remain. In case of significant deviation in dynamic behaviour, the designer should take the quality of the design and simulation models into consideration and redesign the controller.

If the results satisfy, the parameterised control algorithm should be converted towards an, often dedicated, code for implementation in the real-time turbine controller. Therefore, an industrial accepted generic code 'Structured text', as specified in IEC1131-3 [8], is adopted for seamless transfer to the 'system integrating engineers' of the concerning turbine.

The design tool will support the designer in this 'fault sensitive' conversion task by generation of IEC1131-3 compliant code fragments, like definitions of data type declarations, constant settings and parameter assignments from the *control structure modules* and the pertaining *control definition data.*

The 'structured text' or its dedicated conversion to a real-time algorithm should be verified. The design tool will support this by generation of input- and output time series. Comparison between the simulated output series and the algorithm generated output series, both from equal input series, will guarantee proper conversion. Obviously, differences larger than the numerical accuracy are not allowed.

A final and essential iteration of the design procedure should be done by comparison between the real-time performance data (from field measurements) and the expected performance (from simulations).

In case of significant deviation in dynamic behaviour, the designer should take the quality of the design and simulation models into consideration and redesign the controller again.

4. DESIGNER SERVICES

During the design procedure as considered in the previous paragraph, it would be convenient for the designer to have the availability of design tool services.

The control tool provides the following services:

- user input validations on data and signals:
    - checks on completeness, syntax;
    - prevention of double defined data;
- continuous monitoring to screen/file during design:
    - intermediate verification plots;
    - logging messages: progress, warning, abort on error;
- control design figures and data lists for reference reporting;
- mapping between required data and signals for the control structure and the user defined data and signals:
    - checks on missing and superfluous items;
    - dimension checks;
- powerful macros for:
    - retrieval of design and evaluation data;
    - post processing of evaluation data;
- listing and retrieving of design and evaluation data;
- design version control.

In addition, for sake of transparency, repetitive procedures are reduced to simple calls to generalized functions.

5. DESIGN AND EVALUATION MODULES

The design tool comprises design and evaluation modules from research and development on turbine controls. Current (5.1) and foreseen (5.2) modules are listed in the next paragraphs.

5.1 State of the Art

Currently, the design tool comprises generic modules (Matlab m-code) for the design of a PD rotor speed feedback control algorithm, for a variable speed turbine by means of collective pitch actuation and generator torque control. Additional modules are available for improved damping of drive train and tower vibrations. A top-level scheme is given in figure 2.

Both the pitch and torque setpoint can consist of multiple components. These components are then assumed to be independent controllable because of their different operating frequency ranges (band filtering).

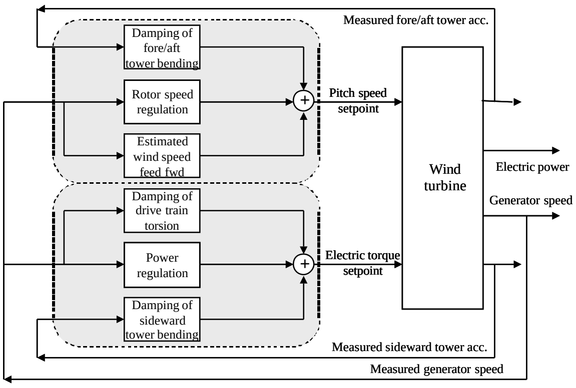

Figure 2: Top level scheme of wind turbine control

To design (synthesis and analysis) this control structure, the following modules are incorporated [1]:

- linear turbine model for control synthesis;
- rotor speed filtering;
- PD rotor speed feedback control (full load);
- pitch angle servo control (partial load);
- two dimensional gain scheduling;
- estimated wind speed feed forward [6];
- dynamic inflow compensation [3;
- electric torque control [11];
- control loop stability assessment;
- non linear control features:
    - inactivity zone;
    - pitch limitation;
    - rotor speed setpoint adaptation;
    - forced rotor speed limitation;
    - control mode setting.

Additional turbine specific modules for load reduction are not incorporated, but also available:

- drive train damping [2];
- tower damping [1] [7];
- crossing of resonance speeds [9].

To evaluate this control structure, the following turbine behaviour is incorporated (see figure 3):

- quasi stationary rotor aerodynamics;
- dynamic inflow [3];
- shaft torsion;
- second order linear tower top translations;
- non linear pitch actuator;
- variable speed generator with E-torque servo;
- delayed measurements;
- rotor effective wind speed;
- tower top equivalent wave forces.

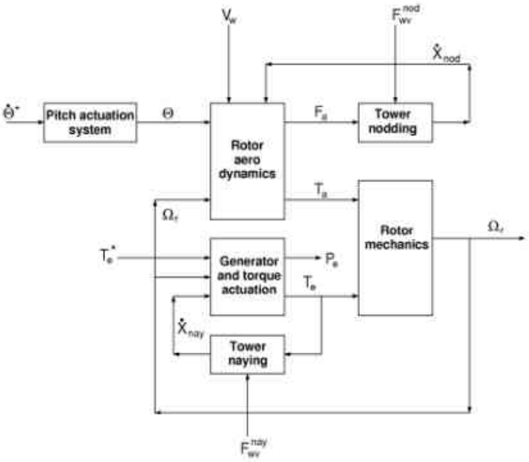

Figure 3: Top level scheme of wind turbine control

The design tool modules as mentioned above, have been applied for development of several industrial control algorithms. The design results have been validated successfully, both in practice by implementation in a modern industrial turbine and by simulations with the aero-elastic code PHATAS.

5.2 Future Developments

Extensions for the design of individual pitch control algorithms are being developed. These algorithms allow for much better load reduction. Points of departure are the integrated linear wind turbine models, which are obtained with the analysis tool 'Frequency Domain Analysis of Offshore Wind Turbines (TURBU)' [4] [5]:

- (reduced) multi body models of coupled dynamics;
- blade effective wind speed;

These developments will be added in short-term to the design tool as design modules and evaluation models.

6. CONCLUSION

The development of a 'Design tool for wind turbine control algorithms' has been resulted in an open source modular design tool within the Matlab environment.

It enables possibilities for wind turbine designers to develop industrial control algorithms and to utilize the benefits of more advanced control solutions.

A proven design procedure takes different stages of wind turbine design into account: from initial control designs based on temporarily turbine specifications via linking to aero-elastic codes for load calculations, through implementation of a final algorithm in a dedicated wind turbine controller.

Designer services are incorporated to operate the design procedure, avoid design failures, and ordering of all the design data, models and design versions.

Currently, the incorporated design and evaluation models are focussed on design of classic 'rotor speed feedback control' for a variable speed and active pitch turbine and have been verified in practice.

More advanced control design modules are within reach as a result of current developments on frequency domain analysis and synthesis of (linearised) turbine models.

ACKNOWLEDGEMENT

ECN Wind Energy has developed the design tool for wind turbine control algorithms. This work is partly funded by Novem, an energy agency of the Dutch Ministry of Economic Affairs.

REFERENCES

[1] Engelen; T.G. van; Hooft, E.L. van der; Schaak, P; "*Development of wind turbine control algorithms for industrial use*"; European Wind Energy Conference 2001 Copenhagen Denmark, proceedings; July 2001.

[2] Engelen; T.G. van; Schaak, P; Lindenburg, C; "*Control for damping the fatigue relevant deformation modes of offshore wind turbines*"; European Wind Energy Conference 2003 Madrid Spain, proceedings; June 2003.

[3] Engelen, T.G. van; "*Dynamic inflow compensation for pitch controlled wind turbines*"; European Wind Energy Conference 2004 London UK; preprint of proceedings; November 2004.

[4] Engelen, T.G. van; Tempel, J. van der; "*Frequency Domain Load Calculation for Wind turbines (TURBU Offshore)*"; European Wind Energy Conference 2004 London UK; preprint of proceedings; November 2004.

[5] Engelen, T.G. van; "Frequency Domain Analysis of Offshore Wind Turbines"; ECN-Wind Memo-04-037; ECN Windenergie, Petten, October 2004.

[6] Hooft, E.L. van der; Engelen, T.G. van; "*Feed forward control of estimated wind speed*"; ECN-C--03-137; ECN Windenergie, Petten, December 2003.

[7] Hooft, E.L. van der; Schaak, P; Engelen, T.G. van; "*Wind turbine control algorithms: DOWEC-F1W1-EH-03-094/0*"; ECN-C--03-111; ECN Windenergie, Petten, December 2003.

[8] Lewis, R.W.; "Programming industrial control systems using IEC1131-3"; IEC control engineering series.

[9] Schaak, P; Corten, G.P.; Hooft, E.L. van der; "*Crossing resonance rotor speeds of wind turbines*"; European Wind Energy Conference 2003 Madrid Spain, proceedings; June 2003.

[10] Schaak, P; Engelen, T.G. van; Hooft, E.L. van der; Wiggelinkhuizen, E.J. "*Development of a Wind turbine control design tool, Phase 1*" (in Dutch); ECN-C--03-133; ECN Windenergie, Petten, December 2003.

[11] Schaak, P; Engelen, T.G. van; "*Torque control for variable speed wind turbines*"; European Wind Energy Conference 2004 London UK; preprint of proceedings; November 2004.

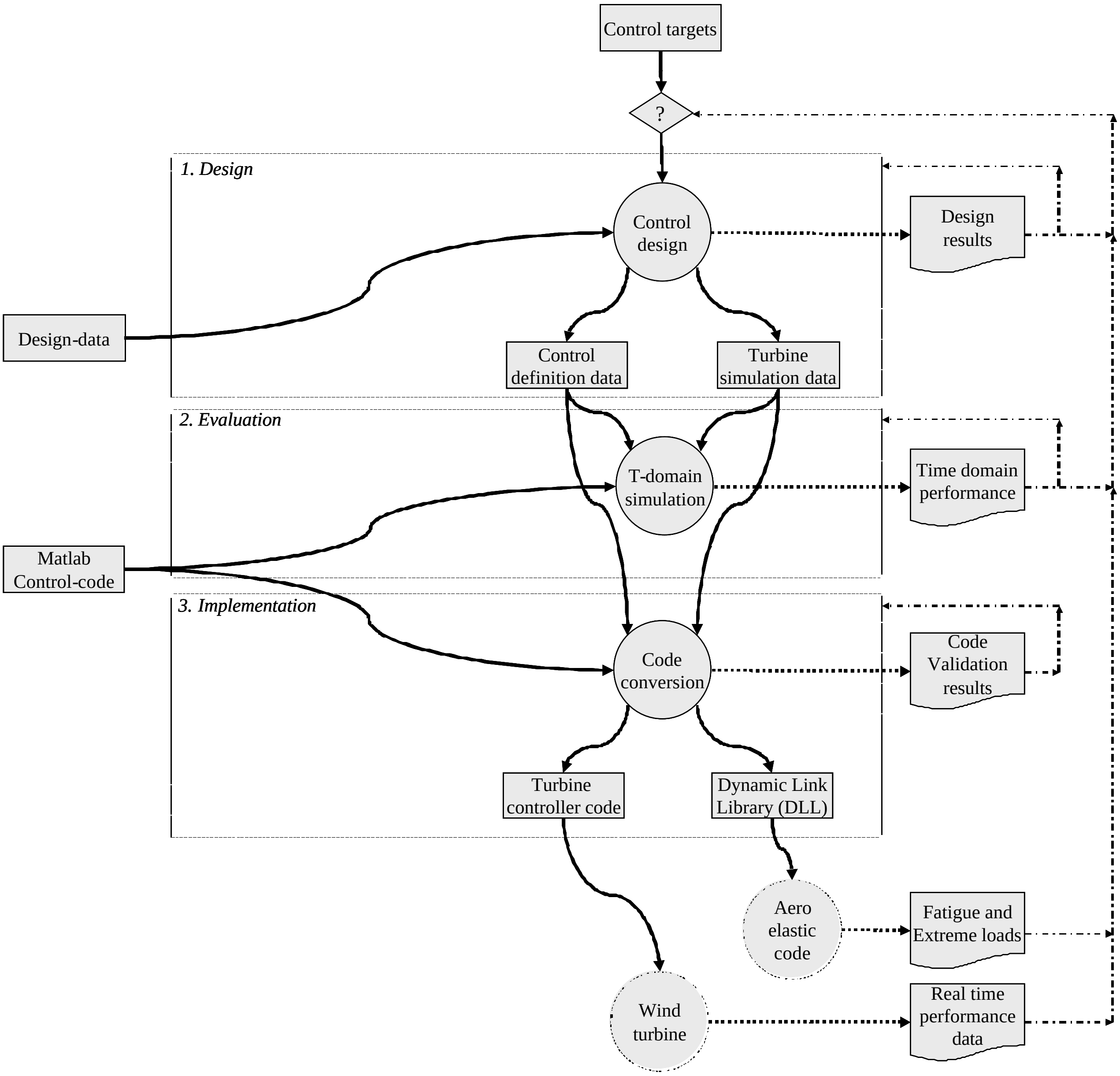

Figure 1: Design procedure for control algorithms.